

NUSC Technical Memorandum 831169
21 November 1983

REFERENCE ONLY



A Program Translator

Marvin J. Goldstein
Donald A. Aker
Computer and Information Services Department

REFERENCE ONLY



Naval Underwater Systems Center
Newport, Rhode Island/New London, Connecticut

Report Documentation Page				Form Approved OMB No. 0704-0188	
Public reporting burden for the collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington Headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to a penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.					
1. REPORT DATE 21 NOV 1983		2. REPORT TYPE Technical Memo		3. DATES COVERED 21-11-1983 to 21-11-1983	
4. TITLE AND SUBTITLE A Program Translator				5a. CONTRACT NUMBER	
				5b. GRANT NUMBER	
				5c. PROGRAM ELEMENT NUMBER	
6. AUTHOR(S) Marvin Goldstein; Donald Aker				5d. PROJECT NUMBER 771Y00	
				5e. TASK NUMBER	
				5f. WORK UNIT NUMBER	
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) Naval Underwater Systems Center, New London, CT, 06320				8. PERFORMING ORGANIZATION REPORT NUMBER TM 831169	
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)				10. SPONSOR/MONITOR'S ACRONYM(S)	
				11. SPONSOR/MONITOR'S REPORT NUMBER(S)	
12. DISTRIBUTION/AVAILABILITY STATEMENT Approved for public release; distribution unlimited					
13. SUPPLEMENTARY NOTES NUWC2015					
14. ABSTRACT In this memorandum, we discuss a computer program that promotes FORTRAN program compilability between computers that have different FORTRAN compilers. The computer program translates VAX FORTRAN structured GO TO-less control elements that are not American National Standard FORTRAN 77 into structured simulated analogs that are compilable by any FORTRAN compiler. This program is intended to complement a recently developed program that promotes program readability.					
15. SUBJECT TERMS FORTRAN					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT Same as Report (SAR)	18. NUMBER OF PAGES 14	19a. NAME OF RESPONSIBLE PERSON
a. REPORT unclassified	b. ABSTRACT unclassified	c. THIS PAGE unclassified			

cy 001

NAVAL UNDERWATER SYSTEMS CENTER
NEW LONDON LABORATORY
NEW LONDON, CONNECTICUT 06320

Technical Memorandum

A PROGRAM TRANSLATOR

Date: 21 November 1983

Prepared by: Marvin J. Goldstein
Marvin J. Goldstein
Computer and Information
Services Department

Donald A. Aker
Donald A. Aker
Computer and Information
Services Department

Approved for public release;
distribution unlimited

ABSTRACT

In this memorandum, we discuss a computer program that promotes FORTRAN program compilability between computers that have different FORTRAN compilers. The computer program translates VAX FORTRAN structured GO TO-less control elements that are not American National Standard FORTRAN 77 into structured simulated analogs that are compilable by any FORTRAN compiler. This program is intended to complement a recently developed program that promotes program readability.

ADMINISTRATIVE INFORMATION

This memorandum was prepared under Job Order No. 771Y00, Special Projects and Studies. The authors are located at the Naval Underwater Systems Center, New London, Connecticut, 06320.

INTRODUCTION

In an earlier memorandum [1], one of the authors discussed a computer program that aids in structuring code entirely in FORTRAN. The computer program is a labor saving tool whose use eliminates effectively the manual effort of formatting code according to programming style conventions [1, 2, 3] that promote FORTRAN program readability. In particular, the computer program eliminates the laborious effort of manually indenting GO TO-less syntactical control elements of structured coding, as well as their simulated transportable analogs, that appear in Figure 1. But the GO TO-less forms, unlike their simulated analogs, are not compilable in general by FORTRAN compilers that are implementations of American National Standard FORTRAN 77 [4], with the exception of the IF THEN ELSE which is part of the standard. For example, although the GO TO-less forms in Figure 1 are compilable by DEC's VAX FORTRAN compiler, they are not compilable by UNIVAC's ASCII FORTRAN compiler. Therefore, if one wishes to promote compilability of FORTRAN programs by different FORTRAN compilers, one should not use the GO TO-less forms.

However, since the GO TO-less forms are easier to code than their simulated analogs, they have been used extensively when supported by a vendor's FORTRAN compiler in spite of the transportability problem that their use presents. Therefore, in order to reduce the reprogramming effort required for FORTRAN programs that contain these forms and will eventually migrate to UNIVAC 1100 series computers, we have written a computer program in PASCAL that will translate the GO TO-less forms in Figure 1 into their simulated transportable forms.

CONTROL STRUCTURES TRANSLATED BY THE TRANSLATOR

If P is a Boolean expression, then the following control structures on the left are translated by the translator into their simulated transportable analogs on the right. New statement labels that are inserted in the simulated forms by the translator are greater than 20000; for example, the translator will generate values greater than 20000 for the statement labels n1 and n2 of the simulated analog of the Block DO WHILE, and for the statement label n1 of the simulated analog of the Block DO n2 WHILE. Note that a new statement label is inserted to implement a simulated form, only if one is not present in the GO TO-less form that can be used.

FIGURE 1

GO TO-less FORMS	SIMULATED (TRANSPORTABLE) FORMS
6100	
6200	
6300	
6400	
6500 GO TO-less FORMS	
6600	
6700	
6800 Block DO WHILE	
6900	
7000 DO WHILE(P)	C DO WHILE(P)
7100 . . .	n1 IF(.NOT.P)GO TO n2
7200
7300
7400 END DO	. . .
7500	GO TO n1
7600	n2 CONTINUE
7700	
7800	
7900 Block DO n2 WHILE	
8000 . . .	
8100 DO n2 WHILE(P)	C DO WHILE(P)
8200 . . .	n1 IF(.NOT.P)GO TO n2
8300
8400
8500 n2 END DO	. . .
8600	GO TO n1
8700	n2 CONTINUE
8800 or	
8900 DO n2 WHILE(P)	
9000 . . .	
9100 . . .	
9200 . . .	
9300 . . .	
9400 n2 CONTINUE	
9500	
9600	
9700 Block DO	
9800 DO J=1,N	DO n1 J=1,N
9900
10000
10100
10200
10300 END DO	n1 CONTINUE
10400	

AN EXAMPLE

Since the computerized translation of GO TO-less forms in a program may upset the order of the program's statement labels, or generate simulated forms that are not indented exactly as they are in Figure 1, resequencing of statement labels, or reindentation of control structures, if desired, can be done easily with the aid of the structured programming tool "CLEAN" described in [1].

For example, consider translating the GO TO-less structures in the program segment in Figure 2. The translator will transform this program segment into the one given in Figure 3, where the statement labels are out of sort and the simulated DO WHILE is not indented exactly as it is in Figure 1. Program CLEAN can be used now to transform the program segment in Figure 3 into the one in Figure 4, where the statement labels are now in order, and the DO WHILE is indented as it is in Figure 1. These transformations can be accomplished on both the VAX 11/780 and UNIVAC 1100/62.

The following sequence of VAX JCL commands will translate Figure 2 into Figure 3, when the translator TRANS in directory [MJG.CLEAN] on nodes 2, 3 and 7 is executed with Figure 2 assigned to the translator's input text file INFILE and Figure 3 to the translator's output text file OUTFILE.

```
$ ASSIGN FIG2.FOR INFILE
$ ASSIGN FIG3.FOR OUTFILE
$ RUN [MJG.CLEAN]TRANS
```

Execution of program CLEAN in directory [MJG.CLEAN] with Figure 3 now assigned to CLEAN's input text file INFILE, and Figure 4 to CLEAN's output text file OUTFILE, resequences statement labels and reindents control structures, using a label increment of 50 and an indentation factor of 3:

```
$ ASSIGN FIG3.FOR INFILE
$ ASSIGN FIG4.FOR OUTFILE
$ RUN [MJG.CLEAN]CLEAN
$ Options: C
$ Label increment: 50
$ Indentation factor: 3
```

The corresponding UNIVAC JCL commands to accomplish these transformations are given below.

Translation of Figure 2 into Figure 3:

```
@ASG,A USER*FILE.
@ASG,T INFILE.
@ASG,T OUTFILE.
@COPY,I USER*FILE.FIG2, INFILE.
@XQT,Y TRANS*PASCAL.TRANS
@COPY,I OUTFILE., USER*FILE.FIG3
```

Transformation of Figure 3 into Figure 4:

```
@COPY,I USER*FILE.FIG3, INFILE.
@XQT,Y CLEAN*PASCAL.UTIL
(User inputs to CLEAN)
@COPY,I OUTFILE., USER*FILE.FIG4
```

When copying FIG2 (or FIG3) into data file INFILE, the cycle number of FIG2 (or FIG3) must be zero.

FIGURE 2

```

16300
16400
16500
16600 C*****
16700 C      AUTHOR: M. J. GOLDSTEIN
16800 C      THIS PROGRAM TESTS THE ACCURACY OF DOUBLE PRECISION PSEUDOINVER-
16900 C      SION ROUTINES. IT PRINTS THE MINIMUM (MINDGT), MAXIMUM (MAXDGT) AND
17000 C      EXPECTED NUMBER OF SIGNIFICANT DIGITS (EXSDGT) IN THE ELEMENTS
17100 C      OF THE COMPUTED PSEUDOINVERSE OF EACH MATRIX A. (THE STANDARD
17200 C*****
17300      DIMENSION A(6,4), AINVT(6,4), G(4,6)
17400      DO WHILE(N.LE.LM)
17500 C
17600 C      GENERATE THE N-TH MATRIX A AND ITS PSEUDO-
17700 C      INVERSE TRANSPOSE AINVT
17800 C
17900      A1=TWO**(2*N)
18000      A1INV=ONE/A1
18100      PRINT 1050,A1,N
18200      DO J=1,4
18300          DO I=1,6
18400              GO TO(50,250,450,450,250,650),I
18500 C          1:
18600 C          50      IF((J.EQ.1).OR.(J.EQ.4))GO TO 100
18700 C                  GO TO 150
18800 C                  THEN
18900 C          100      A(I,J)=A1+A4
19000 C                  AINVT(I,J)=(A1INV+A4INV)/EIGHT
19100 C                  GO TO 200
19200 C          ELSE
19300 C          150      A(I,J)=A1-A4
19400 C                  AINVT(I,J)=(A1INV-A4INV)/EIGHT
19500 C          200      CONTINUE
19600 C                  GO TO 850
19700 C          2,5:
19800 C          250      IF((J.EQ.1).OR.(J.EQ.3))GO TO 300
19900 C                  GO TO 350
20000 C          THEN
20100 C          300      A(I,J)=A2
20200 C                  AINVT(I,J)=A2INV/EIGHT
20300 C                  GO TO 400
20400 C          ELSE
20500 C          350      A(I,J)=-A2
20600 C                  AINVT(I,J)=-A2INV/EIGHT
20700 C          400      CONTINUE
20800 C                  GO TO 850
20900 C          3,4:
21000 C          450      IF((J.EQ.1).OR.(J.EQ.2))GO TO 500
21100 C                  GO TO 550
21200 C          THEN
21300 C          500      A(I,J)=A3
21400 C                  AINVT(I,J)=A3INV/EIGHT
21500 C                  GO TO 600
21600 C          ELSE
21700 C          550      A(I,J)=-A3
21800 C                  AINVT(I,J)=-A3INV/EIGHT
21900 C          600      CONTINUE
22000 C                  GO TO 850
22100 C          6:
22200 C          650      IF((J.EQ.1).OR.(J.EQ.4))GO TO 700
22300 C                  GO TO 750
22400 C
22500 C          THEN
22600 C          700      A(I,J)=A1-A4
22700 C                  AINVT(I,J)=(A1INV-A4INV)/EIGHT
22800 C                  GO TO 800
22900 C          ELSE
23000 C          750      A(I,J)=A1+A4
23100 C                  AINVT(I,J)=(A1INV+A4INV)/EIGHT
23200 C          800      CONTINUE
23300 C          850      CONTINUE
23400 C          END DO
23500 C          END DO
23600 C          N=N+1
23700 C          END DO
23800 C          STOP
23900 C          1050 FORMAT (1H ,19HVALUE OF A1 IS NOW ,D20.12,10H N IS NOW ,I2)
23900 C          END

```

FIGURE 3

```

24100
24200
24300
24400 C*****
24500 C      AUTHOR: M. J. GOLDSTEIN
24600 C      THIS PROGRAM TESTS THE ACCURACY OF DOUBLE PRECISION PSEUDOINVER-
24700 C      SION ROUTINES. IT PRINTS THE MINIMUM (MINDGT), MAXIMUM (MAXDGT) AND
24800 C      EXPECTED NUMBER OF SIGNIFICANT DIGITS (EXSDGT) IN THE ELEMENTS
24900 C      OF THE COMPUTED PSEUDOINVERSE OF EACH MATRIX A. (THE STANDARD
25000 C*****
25100      DIMENSION A(6,4),AINVT(6,4),G(4,6)
25200 C      DO WHILE (N.LE.LM)
25300 49900 IF (.NOT.(N.LE.LM))GO TO 50020
25400 C
25500 C      GENERATE THE N-TH MATRIX A AND ITS PSEUDO-
25600 C      INVERSE TRANSPOSE AINVT
25700 C
25800      A1=TWO**(2*N)
25900      A1INV=ONE/A1
26000      PRINT 1050,A1,N
26100      DO 50010 J =1,4
26200          do 50000 I =1,6
26300              GO TO(50,250,450,450,250,650),I
26400 C              1:
26500 50          IF((J.EQ.1).OR.(J.EQ.4))GO TO 100
26600              GO TO 150
26700 C              THEN
26800 100          A(I,J)=A1+A4
26900              AINVT(I,J)=(A1INV+A4INV)/EIGHT
27000              GO TO 200
27100 C              ELSE
27200 150          A(I,J)=A1-A4
27300              AINVT(I,J)=(A1INV-A4INV)/EIGHT
27400 200          CONTINUE
27500              GO TO 850
27600 C              2,5:
27700 250          IF((J.EQ.1).OR.(J.EQ.3))GO TO 300
27800              GO TO 350
27900 C              THEN
28000 300          A(I,J)=A2
28100              AINVT(I,J)=A2INV/EIGHT
28200              GO TO 400
28300 C              ELSE
28400 350          A(I,J)=-A2
28500              AINVT(I,J)=-A2INV/EIGHT
28600 400          CONTINUE
28700              GO TO 850
28800 C              3,4:
28900 450          IF((J.EQ.1).OR.(J.EQ.2))GO TO 500
29000              GO TO 550
29100 C              THEN
29200 500          A(I,J)=A3
29300              AINVT(I,J)=A3INV/EIGHT
29400              GO TO 600
29500 C              ELSE
29600 550          A(I,J)=-A3
29700              AINVT(I,J)=-A3INV/EIGHT
29800 600          CONTINUE
29900              GO TO 850
30000 C              6:
30100 650          IF((J.EQ.1).OR.(J.EQ.4))GO TO 700
30200              GO TO 750
30300 C              THEN
30400 700          A(I,J)=A1-A4
30500              AINVT(I,J)=(A1INV-A4INV)/EIGHT
30600              GO TO 800
30700 C              ELSE
30800 750          A(I,J)=A1+A4
30900              AINVT(I,J)=(A1INV+A4INV)/EIGHT
31000 800          CONTINUE
31100 850          CONTINUE
31200 50000          CONTINUE
31300 50010          CONTINUE
31400              N=N+1
31500              GO TO 49900
31600 50020          CONTINUE
31700              STOP
31800 1050          FORMAT (1H ,19HVALUE OF A1 IS NOW ,D20.12,10H N IS NOW ,I2)
31900              END

```

FIGURE 4

```

32100
32200
32300
32400 C*****
32500 C      AUTHOR: M. J. GOLDSTEIN
32600 C      THIS PROGRAM TESTS THE ACCURACY OF DOUBLE PRECISION PSEUDOINVER-
32700 C      SION ROUTINES. IT PRINTS THE MINIMUM (MINDGT), MAXIMUM (MAXDGT) AND
32800 C      EXPECTED NUMBER OF SIGNIFICANT DIGITS (EXSDGT) IN THE ELEMENTS
32900 C      OF THE COMPUTED PSEUDOINVERSE OF EACH MATRIX A. (THE STANDARD
33000 C*****
33100      DIMENSION A(6,4), AINVT(6,4), G(4,6)
33200 C      DO WHILE(N.LE.LM)
33300      50      IF(.NOT.(N.LE.LM))GO TO 1050
33400 C
33500 C      GENERATE THE N-TH MATRIX A AND ITS PSEUDO-
33600 C      INVERSE TRANSPOSE AINVT
33700 C
33800      A1=TWO**(2*N)
33900      A1INV=ONE/A1
34000      PRINT 1100,A1,N
34100      DO 1000 J=1,4
34200          DO 950 I=1,6
34300              GO TO(100,300,500,500,300,700),I
34400 C              1:
34500      100          IF((J.EQ.1).OR.(J.EQ.4))GO TO 150
34600                  GO TO 200
34700 C              THEN
34800      150          A(I,J)=A1+A4
34900                  AINVT(I,J)=(A1INV+A4INV)/EIGHT
35000                  GO TO 250
35100 C              ELSE
35200      200          A(I,J)=A1-A4
35300                  AINVT(I,J)=(A1INV-A4INV)/EIGHT
35400      250          CONTINUE
35500                  GO TO 900
35600 C              2,5:
35700      300          IF((J.EQ.1).OR.(J.EQ.3))GO TO 350
35800                  GO TO 400
35900 C              THEN
36000      350          A(I,J)=A2
36100                  AINVT(I,J)=A2INV/EIGHT
36200                  GO TO 450
36300 C              ELSE
36400      400          A(I,J)=-A2
36500                  AINVT(I,J)=-A2INV/EIGHT
36600      450          CONTINUE
36700                  GO TO 900
36800 C              3,4:
36900      500          IF((J.EQ.1).OR.(J.EQ.2))GO TO 550
37000                  GO TO 600
37100 C              THEN
37200      550          A(I,J)=A3
37300                  AINVT(I,J)=A3INV/EIGHT
37400                  GO TO 650
37500 C              ELSE
37600      600          A(I,J)=-A3
37700                  AINVT(I,J)=-A3INV/EIGHT
37800      650          CONTINUE
37900                  GO TO 900
38000 C              6:
38100      700          IF((J.EQ.1).OR.(J.EQ.4))GO TO 750
38200                  GO TO 800
38300 C              THEN
38400      750          A(I,J)=A1-A4
38500                  AINVT(I,J)=(A1INV-A4INV)/EIGHT
38600                  GO TO 850
38700 C              ELSE
38800      800          A(I,J)=A1+A4
38900                  AINVT(I,J)=(A1INV+A4INV)/EIGHT
39000      850          CONTINUE
39100      900          CONTINUE
39200      950          CONTINUE
39300      1000         CONTINUE
39400                  N=N+1
39500                  GO TO 50
39600      1050         CONTINUE
39700                  STOP
39800      1100         FORMAT (1H ,19HVALUE OF A1 IS NOW ,D20.12,10H N IS NOW ,I2)
39900                  END

```

PROGRAM FEATURES

The following program features should be emphasized:

1. Each FORTRAN module processed by the program translator must terminate on the FORTRAN END statement.
2. The translator will process FORTRAN modules that use either upper or lower case characters.
3. The translator will process all program modules in a FORTRAN file so that a FORTRAN file containing more than one FORTRAN module may be assigned to the text file INFILE.
4. Only the GO TO-less structures in Figure 1 are modified by the translator. All other FORTRAN statements remain unchanged! Therefore, FORTRAN modules that do not use any of the GO TO-less structures in Figure 1 are left unchanged by the translator.
5. Modified statements occupy the same print positions as the statements that they replace; however, new statements that are added (conditional and explicit transfers of control) are not indented to conform with the indentation convention shown in Figure 1.
6. New statement labels generated by the translator have values greater than 20000, so that, in general, they will not interfere with existing statement labels.
7. To complement the program translator, program "CLEAN" can be used to resequence statement labels and reindent control structures!

OVERVIEW OF TRANSLATION ALGORITHM

The translation algorithm is based on the observation that the GO TO-less structures in Figure 1 can be translated into their simulated analogs by using push-down stacks. Essentially the line numbers of statements that introduce GO TO-less structures are placed on a push-down stack when they are encountered. When an END DO (structure terminator) is encountered, however, the stack is "popped" and the structure terminator is paired with the line number at which the structure is introduced.

For example, consider the following simple nest of GO TO-less structures:

```

line 1      DO I = 1, N
line 2      DO J = 1, N
              C(I,J) = 0.0
            END DO
          END DO

```

When a DO line is identified, its line number (line i) is placed on a push-down stack, so that immediately after the second DO has been processed, the push-down stack for our simple nest contains

```

line 2      <--- top of stack
line 1

```

When an END DO is identified, it is converted to a labeled CONTINUE statement, the line number at the top of the stack is removed and placed on an auxiliary stack along with the label, n(3-i), of the CONTINUE statement. This properly pairs in the auxiliary stack the line number introducing a GO TO-less structure with the label assigned to the structure terminator. Processing the code from beginning to end in this manner, the translator generates the auxiliary stack:

Auxiliary stack:

```

line 1, n2  <--- top of stack
line 2, n1

```

and the partially translated code:

```

line 1      DO I = 1, N
line 2      DO J = 1, N
              C(I,J) = 0
n1          CONTINUE
n2          CONTINUE

```

Now reading the partially translated code from beginning to end, when a line number is encountered that matches the line number on the top of the auxiliary stack, the translator inserts the stack label in the corresponding line of FORTRAN and then "pops" the stack. This procedure is repeated until the auxiliary stack is empty. The final translated code is

```

          DO n2 I = 1, N
            DO n1 J = 1, N
              C(I,J) = 0.0
n1          CONTINUE
n2          CONTINUE

```

SUMMARY

A computer program is available that translates VAX FORTRAN structured GO TO-less control elements that are not American National Standard FORTRAN 77 into structured simulated analogs that are compilable by any FORTRAN compiler. This capability reduces the manual reprogramming effort required to successfully recompile VAX FORTRAN programs on other computer systems with FORTRAN compilers that will not compile these non-standard control elements. This capability, like program "CLEAN" [1], is being provided to NUSC/NET users as a software development aid in producing more maintainable FORTRAN programs.

REFERENCES

1. Marvin J. Goldstein and John Lawson, Jr., "A New Program Aid in Producing Structured FORTRAN Programs," NUSC TM No. 821162, 9 November 1982.
2. Marvin J. Goldstein, David K. Anderson and Drew Drinkard, "A PASCAL Program for Producing Structured FORTRAN Programs," NUSC TM No. 811153, 22 October 1981.
3. M. J. Goldstein and John Lawson, Jr., "An Example of Quality Mathematical Software," NUSC TM No. 811044, 15 April 1981.
4. American National Standard Programming Language FORTRAN, ANSI X3.9 1977, American National Standards Institute, New York, NY, 1978.

A PROGRAM TRANSLATOR

Marvin J. Goldstein, Computer and Information Services Dept.

Donald Aker, Computer and Information Services Dept.

TM No. 831169

21 November 1983

UNCLASSIFIED

DISTRIBUTION LIST

EXTERNAL

Mr. I. L. Avrunin
David Taylor Naval Ship
Research & Development
Center
Headquarters
Bethesda, MD 20084

Mr. James Dooley
Naval Data Automation
Command
Washington Navy Yard
Washington, DC 20374

Mr. John Baird
NOSC
San Diego, CA 92152

Ms. Mary Ann Engelbert
Naval Data Automation
Command
Washington Navy Yard
Washington, DC 20374

Ms. Fran Kazlauskis
Naval Data Automation
Command
Washington Navy Yard
Washington, DC 20374

Dr. Martin Schultz
Research Center for Scientific Comp.
Yale Univ.
P. O. Box 2158 Yale Station
New Haven, CT 06520

Mr. Ken Kunec
Naval Data Automation
Command
Washington Navy Yard
Washington, DC 20374

Mr. Al Poulin
Naval Intelligence Command
Headquarters
4600 Silver Hill Road
Washington, DC 20389

Ms. Becky Swales
Naval Data Automation
Command
Headquarters
4600 Silver Hill Road
Washington, DC 20389

Mr. Larry Williams
Naval Intelligence
Command
Headquarters
4600 Silver Hill Road
Washington, DC 20389

INTERNAL

CODE 00	CAPT Ailes	Code 33	J. J. Higgs	Code 401	B. A. Radley
01	E. L. Messere		I. B. Cohen		A. Y. Shigematsu
01A	John Keegan		D. W. Counsellor	402	M. A. Tucchio
039	Dr. D. M. Viccone		D. A. Fingerman	4111	M. Berger
10	Dr. W. A. VonWinkle		R. J. Trembley	4331	Samuel Horvitz
101	Dr. E. Eby		J. E. Miller		B. L. Antrim
02	D. C. Kindilien		Dr. H. F. Dwyer		S. B. Walsh
0211	R. H. Bernier		Dr. C. Carter	5202	Stephan Schady
02	NL Library (3 copies)		Peter Stahl	60	Dr. D. Bordelon
021311	NPT Library		Dr. M. B. Moffett	70	Dr. Jeffrey Cohen
02122	Distribution Center (10 Copies)		C. J. Becker	701	Cort R. DeVoe
20	W. L. Clearwaters		F. C. Gannon		R. Wilson Npt
234	R. J. Conley		R. J. MacDonald		R. Streit
32	A. Lesick		M. H. Richardson		D. McCue
	Dr. N. L. Owsley		J. Saikowski	71	J. Babiec Npt
	W. D. T. Porter		G. Botseas		G. Elias Npt
	H. Axtell		Dr. D. Lee		M. Goldstein (15 copies)
	R. C. Cox		J. Nordquist		D. Drinkard
	James Pearson		E. R. Robinson		D. Aker (15 copies)
	L. C. Ng		H. Sternberg		N. Sulinski
	J. V. Sanchis		Dr. David Wood		W. Babson
	D. M. Finney		P. N. Mikhalevsky		E. Montavon
	B. Helme, Jr.	34	Michael Fetcher		J. Sikorski
	Dr. C. H. Sherman		J. R. Katan		Louis Ledoux Npt
	W. Fox		Dr. P. E. Fessenden		R. Johnson
	B. G. Buehler		C. N. Flemings		H. Eley
	A. W. Ellenthorpe		K. F. Hafner		Cynthia Borges Npt
	Dr. Peter G. Cable		John Casey		Marcel Nadeau Npt
	Dr. S. Ko		Anthony Bruno		John Ventura Npt
	D. T. Porter	35	D. Dixon		Phoebe Liu Npt
	W. H. Wharton		Tom Conrad		John Sabulis Npt
	T. Anderson	36	L. Cabral	72	John Lawson, Jr.
	J. Ionata		R. P. Eidimtas		Gordon Daglieri Npt
	M. Kuuznitz		H. A. Rosen		S. Schneller Npt
	J. Gannon		R. V. Cherry		J. Auwood
	C. Bowman		W. J. Ryan		T. Wheeler
	P. R. Miner		J. Griffin		R. Warren Npt
	Rosemary Molino		D. G. Blundell		P. Breslin Npt
	W. A. Clearwaters		M. P. Lydon		A. Alfiero
	Jose Munoz		S. I. Wax	73	R. Clark
	J. Shores		S. E. Ashton		J. Gribbin
	Dr. John Hammer		Raymond McMahon		D. Quigley
	R. Leask	37	S. Meyers		S. Capizzano
	Russell Christman	38	C. M. Curtis		J. MacDonald
33	Dr A. H. Nuttal	401	J. E. Sims		R. Pingree
	J. B. Paniszczyn		A. D. Carlson		R. Cote
	Dr. F. R. DiNapoli		J. H. Clark		R. Forget
	E. P. Jensen		Dr. A. J. Kalinowski	74	A. Sullivan
	E. G. Kanabis		Dr. R. G. Kasper		J. Surdo
	S. C. Gerengher		R. R. Manstan		M. Lee
	J. Gregor		E. L. McLaughlin		C. Brockway
	W. A. Goldman		R. S. Munn		M. Becker
	P. M. Anchors		C. W. Nebelung		L. Doyle
			Dr. J. S. Patel		J. Koonce
					J. Pappadia
					A. Levander
					Norman Dube Npt
					A. Blau Npt
					D. Stanhope Npt
					R. Hoy Npt
					N. Bradbury Npt